ARMY RESEARCH LABORATORY

Ultra-Wideband Harmonic Radar for Locating Radio-Frequency Electronics

by Gregory J Mazzaro, Kyle A Gallagher, Albert R Owens, Kelly D Sherbondy, and Ram M Narayanan

ARL-TR-7256 **March 2015**

NOTICES

Disclaimers

Army Research Laboratory
Adelphi, MD 20783-1138

ARL-TR-7256 **March 2015**

Ultra-Wideband Harmonic Radar for Locating Radio-Frequency Electronics

Gregory J Mazzaro, Kyle A Gallagher, Albert R Owens,
Kelly D Sherbondy, and Ram M Narayanan
Sensors and Electron Devices Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* March 2015	**2. REPORT TYPE** Final	**3. DATES COVERED (From - To)** 06/2014–07/2014
4. TITLE AND SUBTITLE Ultra-Wideband Harmonic Radar for Locating Radio-Frequency Electronics		**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Gregory J Mazzaro, Kyle A Gallagher, Albert R Owens, Kelly D Sherbondy, and Ram M Narayanan		**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-SER-U 2800 Powder Mill Road Adelphi, MD 20783-1138		**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-7256
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report describes a type of nonlinear radar that detects and locates nonlinear (electronic) targets using ultra-wideband transmission and harmonic reception. Data are presented for a successful test of an experimental harmonic radar at transmit frequencies between 700 and 900 MHz and an output power of 2 W. Detection is accomplished by measuring a signal reflected from the target at double the transmit frequency. Ranging is accomplished by sweeping the transmit frequencies across an ultra-wide bandwidth, capturing the amplitude and phase of the harmonic return signal, and computing an inverse Fourier transform of the wideband harmonic response. Detection and ranging of 3 nonlinear targets are demonstrated at standoff distances of 6 and 12 ft.

15. SUBJECT TERMS

nonlinear, harmonic, radar, ultra-wideband

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
A. Report	**b. ABSTRACT**	**c. THIS PAGE**			Kelly D Sherbondy
Unclassified	Unclassified	Unclassified	UU	46	**19b. TELEPHONE NUMBER** (*Include area code*) (301) 394-2533

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

Nonlinear radar is well suited for the detection of electronic devices, typically those containing semiconductors and whose traditional (linear) radar cross sections are very low owing to their thin geometric profile. Nonlinear radar provides high clutter rejection,[1] but requires two major tradeoffs: 1) the power-on-target required to generate a signal-to-noise ratio comparable to linear radar is much higher than that of linear radar,[2] and 2) the radar must capture weak target responses in the presence of strong transmitted probes.[3]

The nonlinear radar studied in this report transmits a single frequency at a time, denoted f_0, and receives twice this value, $2f_0$; thus, the radar is harmonic. Prior work has focused on transmitting power at f_0 sufficient to generate a harmonic target response while minimizing the amount of transmitter-generated harmonic that couples to the receiver.[4] More recent work has focused on amplifying a weak target response at $2f_0$ while minimizing the amount of receiver-generated harmonic that masks this response.[5] This report demonstrates over-the-air detection and ranging of nonlinear targets using an ultra-wideband transmission and harmonic reception.

2. Background

The basic architecture of a harmonic radar is shown in Fig. 1. The radar transmits a single frequency f_0, electromagnetic energy at this frequency arrives at the target, the target captures and reradiates electromagnetic energy at harmonics of the original frequency, and the radar listens for a particular harmonic, in this case, $2f_0$. Reception of the harmonic indicates detection of an electromagnetically nonlinear target. Typically, because nonlinear target responses are so weak, filtering and amplification in the receive chain are separated into multiple stages.[6–9]

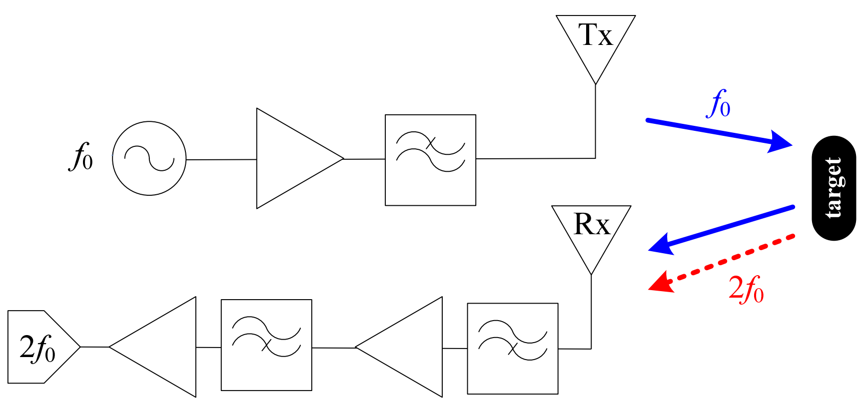

Fig. 1 A harmonic radar that transmits at the fundamental frequency f_0 and receives at the harmonic $2f_0$. Transmitter = "Tx", receiver = "Rx".

In current literature, harmonic radar is typically narrowband.[1,6–15] For narrowband systems, target detection and classification are possible, but unless a collection of antennas or a mobile platform is implemented,[16] range information is not available. Ranging of nonlinear targets is possible for a single stationary pair of antennas, though, if harmonic responses are collected over an ultra-wide bandwidth.[17]

By sweeping the transmit frequency f_0 with known amplitude and phase across an ultra-wide bandwidth and recording the amplitude and phase of the harmonic return signal at each $2f_0$, a nonlinear frequency response of the radar environment is constructed. An inverse Fourier transform of this nonlinear frequency response reveals the range to the target. Presented in this report is an extension of a bench-top, wireline experiment that we recently developed[17] to a wireless, over-the-air data collection using a single ultra-wideband antenna for transmission and reception. Detection and ranging are demonstrated using 3 electromagnetically nonlinear targets.

3. Experiment

A block diagram of the experimental ultra-wideband harmonic radar constructed at the Adelphi Laboratory Center (Building 507) is shown in Fig. 2. The power budget for the signal transmitted at f_0 and received at $2f_0$ is given in Fig. 3. It should be noted that this power budget is approximate, because the gain/loss of each component/cable depends on f_0.

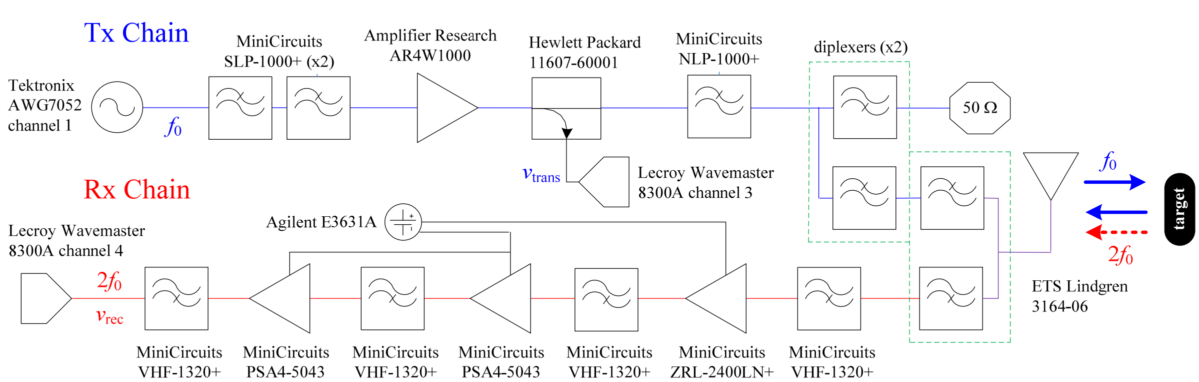

Fig. 2 Block diagram of an experimental ultra-wideband harmonic radar

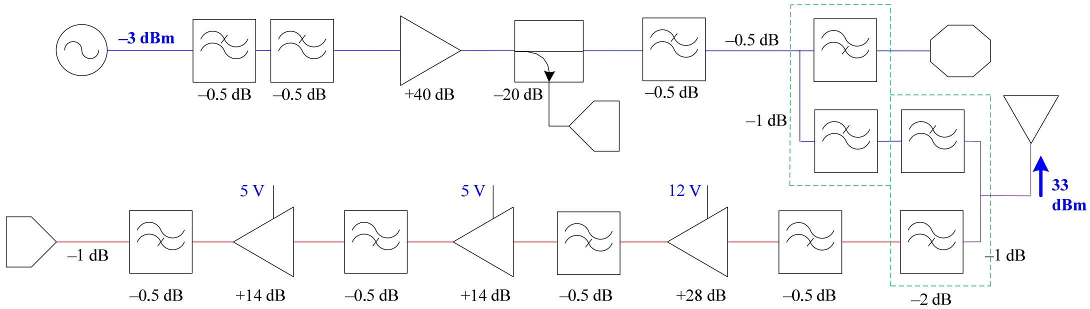

Fig. 3 Approximate power budget for the experimental harmonic radar

The signal source is the Tektronix AWG7052 arbitrary waveform generator (AWG). It outputs signals in the frequency range f_0 = 700 to 900 MHz, at a power of –3 dBm (0.5 mW). The remainder of the transmit chain is intended to strongly amplify f_0 and strongly attenuate $2f_0$. The AWG signal is filtered to remove harmonics using 2 cascaded MiniCircuits SLP-1000+ lowpass filters. The relatively weak signal at f_0 is increased by 40 dB by the Amplifier Research AR4W1000 power amplifier. The output of the AR4W1000 is filtered by a MiniCircuits NLP-1000+ lowpass filter. On its way to the diplexer pair, the transmitted signal is sampled using the Hewlett Packard 11607-60001 (20 dB) directional coupler. The sampled signal is captured by 1 channel of the Lecroy Wavemaster 8300A oscilloscope. Most of the transmitted power passes through the coupler (with negligible loss) to the input of the first diplexer. Approximately 33 dBm (2 W) reaches the antenna to be transmitted (at each frequency).

Two diplexers are placed between the transceiver and the radar antenna. The lowpass output of the first diplexer is connected to the lowpass output of the second. The highpass output of the first diplexer is connected to a matched load (50 Ω), which functions as a fourth filter in the transmit path. Unlike the 3 MiniCircuits lowpass filters, however, the diplexer terminates (absorbs) harmonics, i.e., it does not reflect them back into the transmitter where they may couple into the receiver. The signal that passes through the diplexers — in the forward direction — is sent to the ETS Lindgren 3164-06 horn antenna (“vertical” port), and this antenna transmits f_0 into the air. The harmonics generated by the target are received by the same antenna (on the same polarization).

Because the highpass output of the second diplexer is connected to the receive chain, only harmonics pass through the diplexers in the reverse direction. Therefore, the diplexers, connected back-to-back, 1) provide lowpass filtering in the forward direction, 2) provide highpass filtering in the reverse direction, and 3) allow a single antenna to transmit f_0 and receive $2f_0$. It should be emphasized that the Tx and Rx antennas need not be the same antenna. (If the Tx and Rx antennas were separated, the lowpass output of the first diplexer would feed directly into the Tx antenna instead of passing through the second diplexer, and the lowpass output of the second diplexer would be terminated in a matched load. The transceiver configuration would otherwise remain the same.)

The receive chain (beyond the diplexers) consists of 4 filters and 3 amplifiers, intended to strongly attenuate f_0, strongly amplify $2f_0$, and ultimately capture the amplitude and phase of signals in the frequency range $2f_0$ = 1400 to 1800 MHz. Each of the 4 filters is a MiniCircuits VHF-1320+ highpass filter. The first low-noise amplifier is the MiniCircuits ZRL-2400LN+; the following 2 amplifiers are MiniCircuits PSA4-5043 evaluation boards, powered by the Agilent E3631A (triple-output DC power supply). The received signal is captured by a second channel of the Lecroy Wavemaster 8300A oscilloscope.

Capturing the transmit and received signal simultaneously is necessary to determine the phase difference between the outgoing f_0 and incoming $2f_0$. This phase difference, captured across a wide band of frequencies, is critical to determining range-to-target (as calculated in Section 4).

S-parameter measurements for portions of the radar were recorded using the Agilent N9923A ("FieldFox") portable vector network analyzer. Some of the S-parameters are reported in this section, but most are reported in Appendix A. The magnitude of the transmission S-parameter (S_{21}) for both the transmitter (between the AWG and the antenna port) and the receiver (between the antenna port and the oscilloscope) are given in Fig. 4. The net effect of the lowpass filters, power amplifier, and diplexers (in the forward direction) is to amplify f_0 by about 36 dB and attenuate $2f_0$ substantially. The net effect of the diplexers (in the reverse direction), highpass filters, and low-noise amplifiers is to amplify $2f_0$ by about 50 dB and attenuate f_0 substantially.

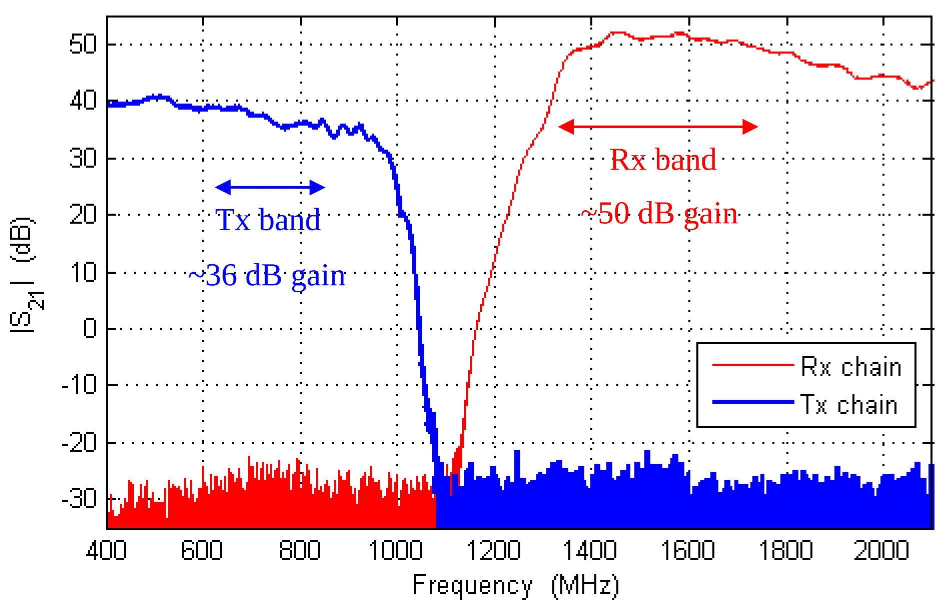

Fig. 4 Overall measured $|S_{21}|$ for the transmit and receive chains

Measured S-parameters for the diplexer pair are given in Fig. 5. The input from the transmitter is denoted Port 1, the output to the antenna is denoted Port 2, and the output to the receiver is denoted Port 3. Signals between f_0 = 700 to 900 MHz pass from Port 1 to Port 2 with <2 dB of loss ($|S_{21}| > -2$ dB in this range), and less than 10% of the energy from signals between $2f_0$ = 1400 to 1800 MHz reflect from the diplexer back into the transmitter ($|S_{11}| < -10$ dB in this range). Signals between $2f_0$ = 1400 to 1800 MHz pass from Port 2 to Port 3 with <1 dB of loss ($|S_{32}| > -1$ dB in this range).

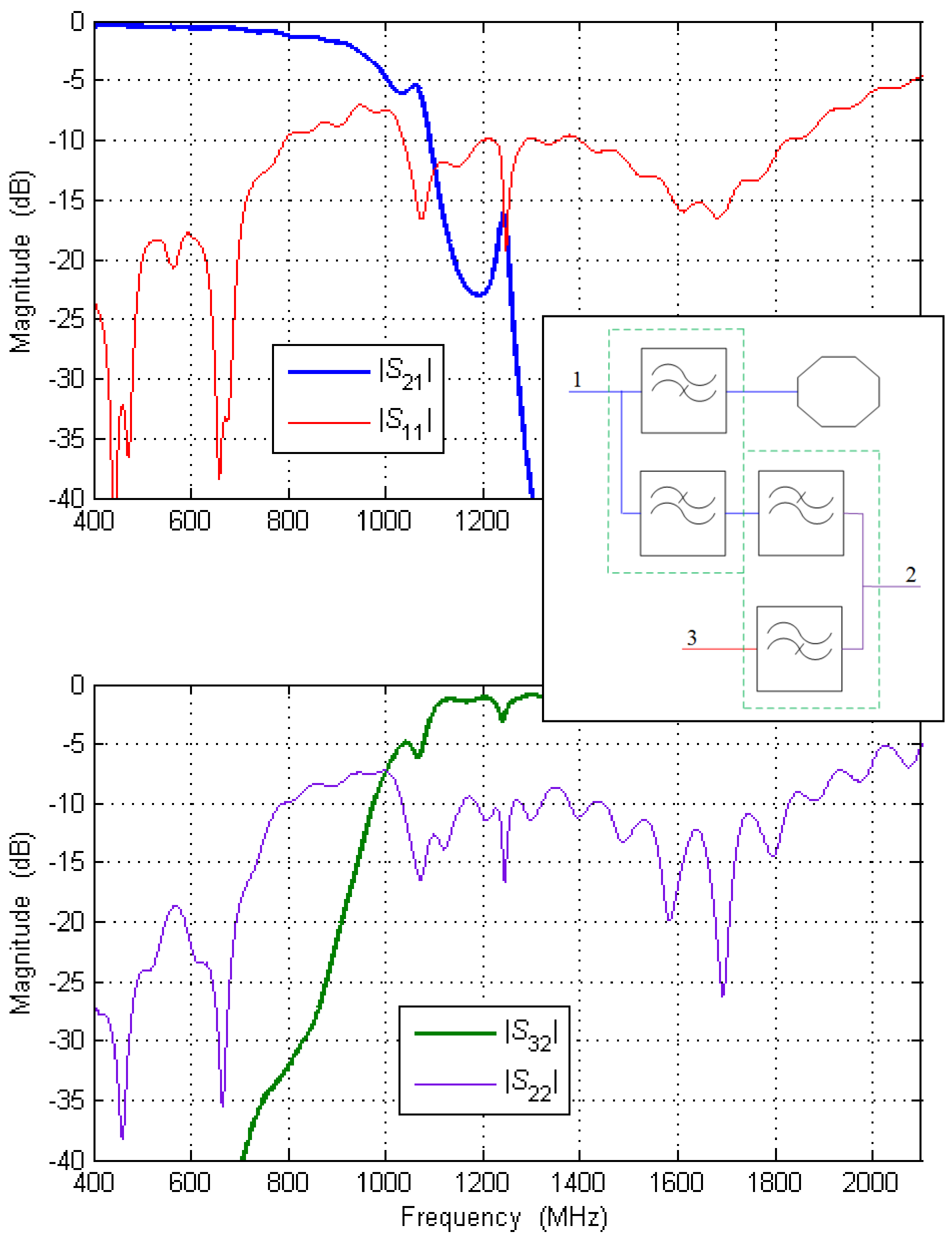

Fig. 5 Measured S-parameters for the pair of (back-to-back) diplexers

Pictures of the experimental harmonic radar are shown in Figs. 6–11. Figure 6 is a picture of the signal generation and capture equipment, sitting on a cart and connected to the Tx/Rx antenna by a single 6-ft SMA cable. Figure 7 is a closer view of the Tx/Rx antenna and the target location (atop an empty cardboard box, 12 ft away from the round metal back-plane of the antenna). Figure 8 is a picture of the Panasonic ToughBook laptop, running Matlab, used to collect and process the radar data. Figure 9 is a closer view of the directional coupler and the receive chain. Although the diplexers are visible in Fig. 9, a much closer view of them is shown in Fig. 10. Figure 11 is a picture of the target set — 3 different nonlinear radio-frequency electronic targets (a connectorized handheld radio, an unpowered low-noise amplifier, and a disconnected radio-frequency mixer) and 2 different antennas (6 inches long vs. 16 inches long).

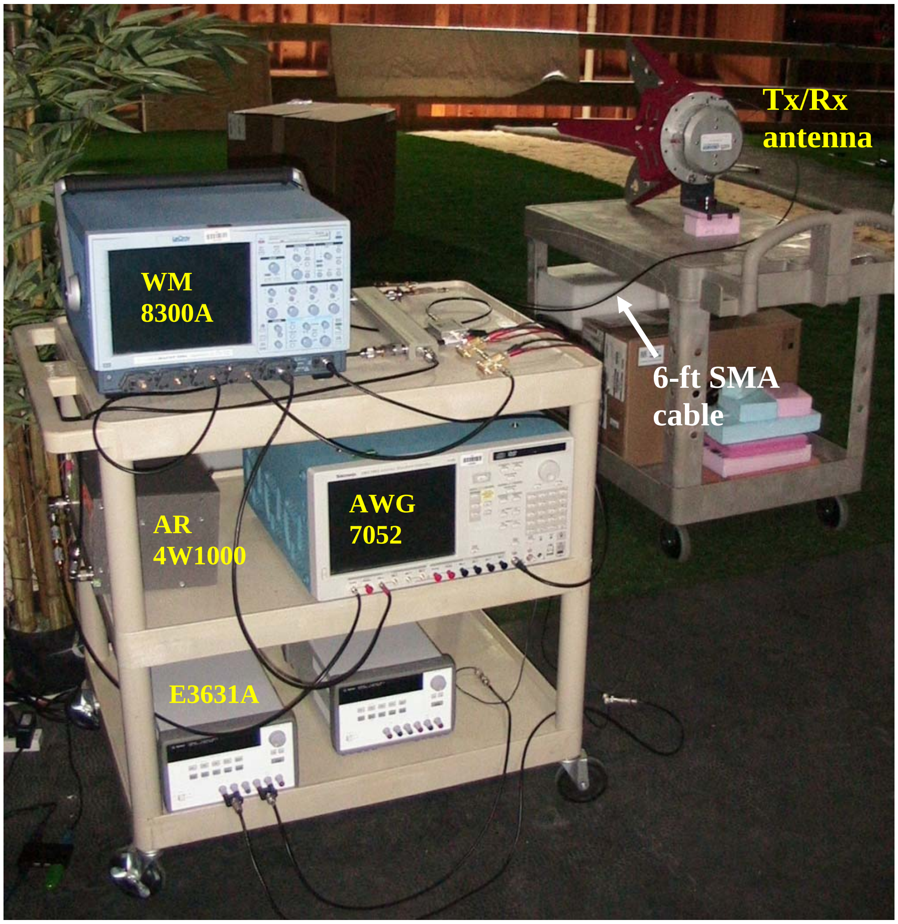

Fig. 6 Picture of the experimental harmonic radar: signal capture and generation cart

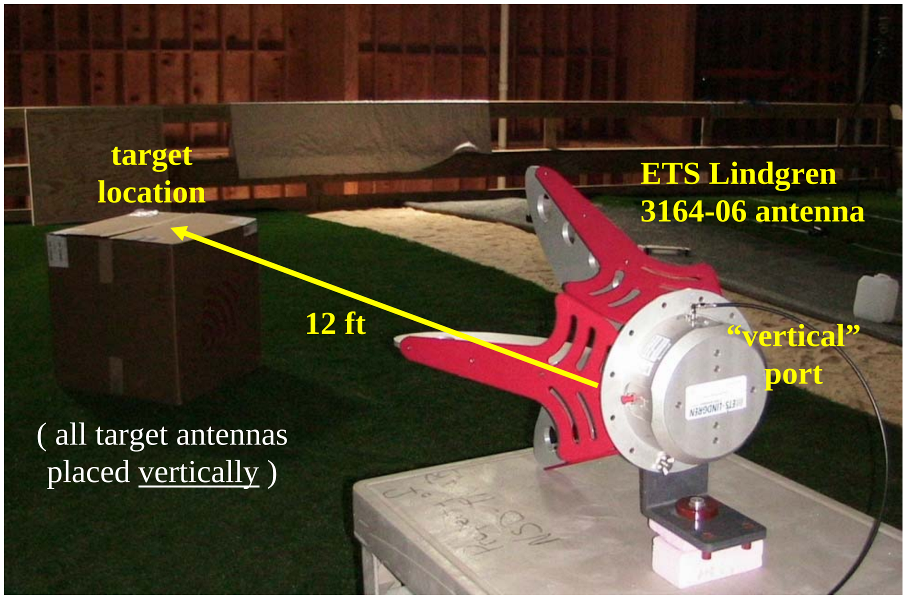

Fig. 7 Picture of the Tx/Rx antenna and target emplacement: Adelphi Laboratory Center, Building 507, “sandbox” area.

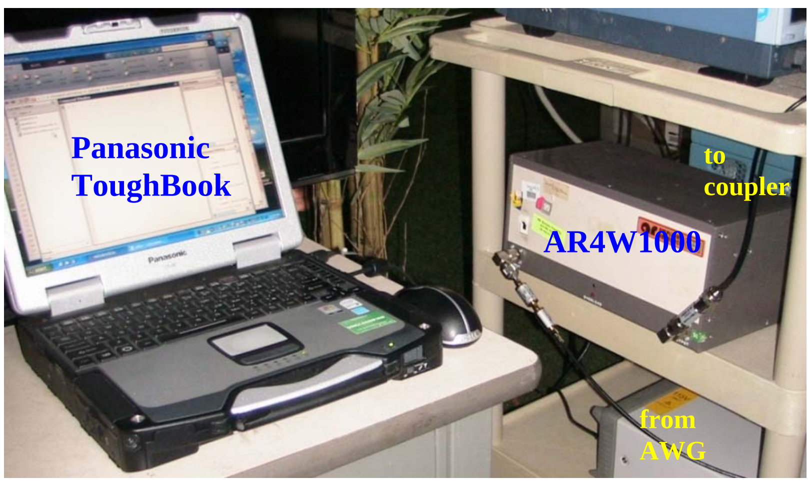

Fig. 8 Picture of the laptop controller and the power amplifier

Fig. 9 Picture of the directional coupler, diplexers, and receive chain

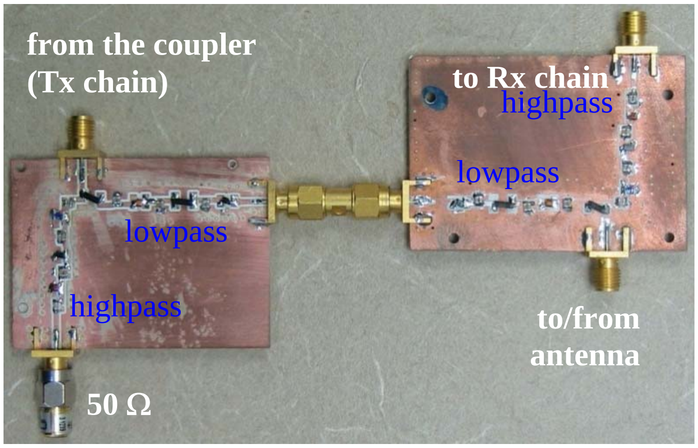

Fig. 10 Close-up picture of the diplexers: discrete components are surface-mounted on a Rogers 4350B substrate

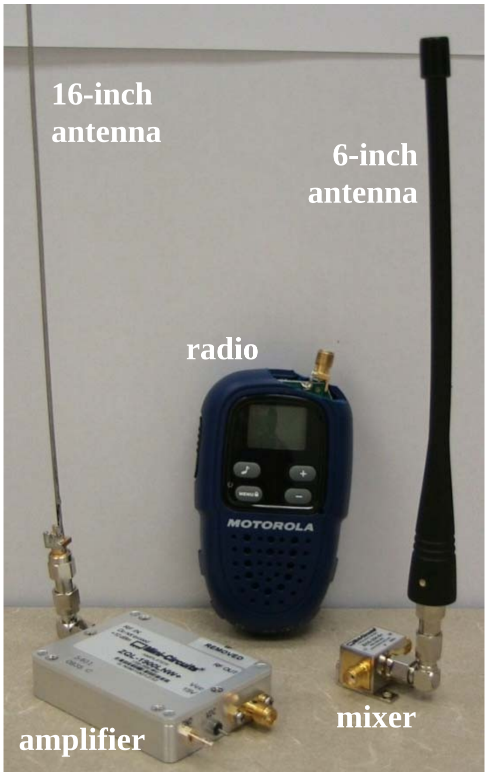

Fig. 11 Picture of the target set: MiniCircuits ZQL-1900LNW+ amplifier (not powered), Motorola FV300 radio (antenna port), MiniCircuits ZX05-20H-S+ mixer (intermediate frequency [IF] port)

4. Theory and Data Processing

For this experiment, the waveform implemented to collect data across the frequency range f_0 = 700 MHz to f_0 = 900 MHz was a linear chirp. If the transmitted waveform is a constant-amplitude sinusoid whose frequency sweeps linearly from f_{start} to f_{end}, the voltage applied to the transmit antenna may be represented by the following form:

$$v_{\mathrm{trans}}(t) = V_0 \cos[2\pi \cdot f_0(t) \cdot t] \qquad f_0(t) = f_{\mathrm{start}} + \frac{f_{\mathrm{end}} - f_{\mathrm{start}}}{T_{\mathrm{env}}} t \tag{1}$$

T_{env} is the length of the transmitted pulse (while it is active). If the radar receiver is designed such that a single harmonic M is selected for reception, the target response — measured back at the radar — takes the form

$$v_{\mathrm{rec}}(t) = |V_M| \cos[2\pi \cdot M \cdot f_0(t) \cdot (t - 2\tau) + \phi\{V_M\}] \tag{2}$$

where 2τ is the time-of-flight of the radar wave from the transmitter, to the target, and back to the receiver. By capturing the amplitude and phase of the harmonic response at each frequency, the following target response may be constructed:

$$\tilde{H}_{\mathrm{NL}}(f) = \frac{\tilde{V}_{\mathrm{rec}}}{\tilde{V}_{\mathrm{trans}}} = \begin{cases} |V_M| V_0^{M-1} e^{-j2\pi \cdot f\tau} & , \quad |f - f_{\mathrm{tgt}}| \le B_{\mathrm{tgt}}/2 \\ 0 & , \quad |f - f_{\mathrm{tgt}}| > B_{\mathrm{tgt}}/2 \end{cases} \tag{3}$$

H_{NL} is the nonlinear (harmonic) frequency response of the target. The response is nonzero over a finite bandwidth B_{tgt} (limited either by the target response or by the transmit frequency set), centered at f_{tgt}. The phase shift $\phi\{V_M\}$ is assumed constant over the band-of-interest and is therefore suppressed. Under the assumption that $|V_M|$ is also constant over the target response band, the inverse Fourier transform of H_{NL} is

$$h_{\mathrm{NL}}(t) = \frac{\sin\{B_{\mathrm{tgt}}(t-2\tau)\}}{\pi(t-2\tau)} |V_M| V_0^{M-1} e^{j(2\pi)(f_{\mathrm{tgt}})(t-2\tau)} \tag{4}$$

which is a sinc function, delayed by twice the time-of-flight of the received radar wave, and scaled by the harmonic amplitude response of the target. Plotted against $d = u_p t/2$ instead of time, where u_p is the velocity of propagation of the radar wave, the peak of h_{NL} indicates the range to the nonlinear target. For a radar environment that is purely linear (i.e., when no nonlinear target is present), $h_{\mathrm{NL}} = 0$ for all values of t.

5. Data Capture and Results

Instrument control for this experiment was performed using Matlab (version R2013b, including the Instrument Control toolbox, over a general purpose interface bus [GPIB]-universal serial bus [USB] interface). A screenshot of the graphical user interface (GUI) created to upload waveforms to the Tektronix waveform generator and download/process data from the Lecroy oscilloscope is given in Fig. 12. For this experiment, a single waveform (whose parameters are illustrated) was uploaded to the AWG. For each target and distance away from the Tx/Rx antenna, a pair of data traces was recorded (v_{trans} vs. t_{trans} and v_{rec} vs. t_{rec}, inside each “MAT” file) and the harmonic impulse response h_{NL} was computed. The resulting h_{NL} traces are shown in Figs. 13–16.

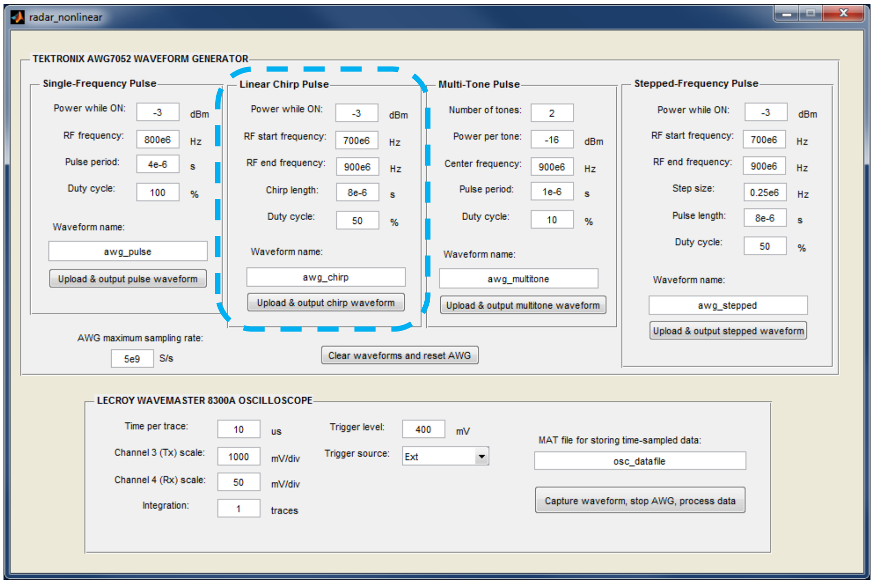

Fig. 12   GUI for harmonic radar instrument control

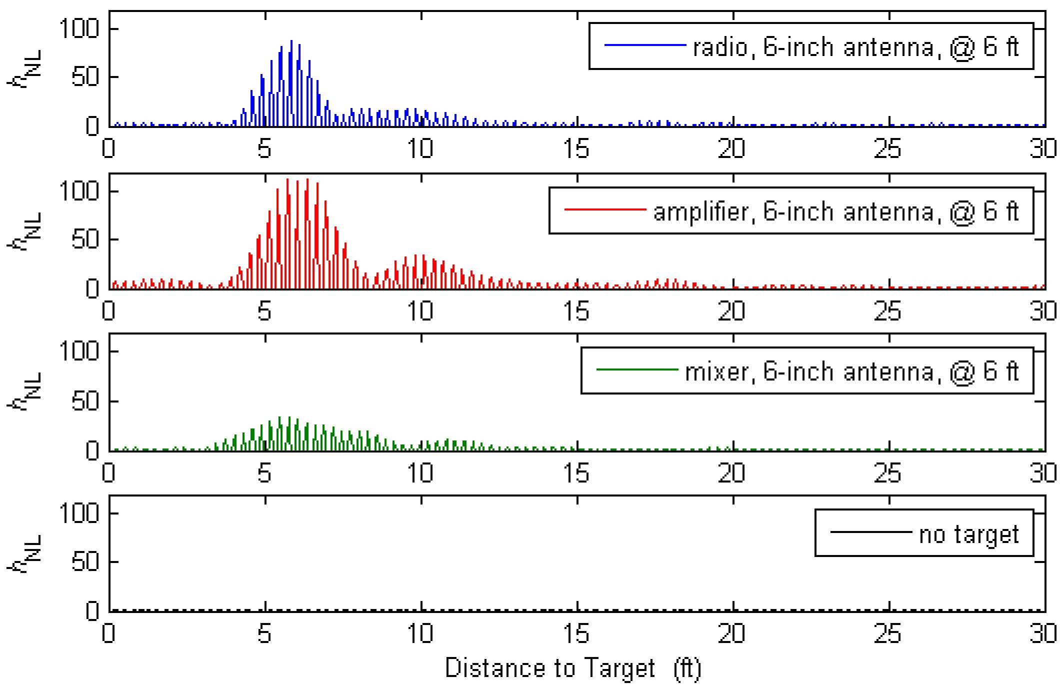

Fig. 13   Data processed for 3 targets with 6-inch antennas, placed 6 ft away from the radar antenna

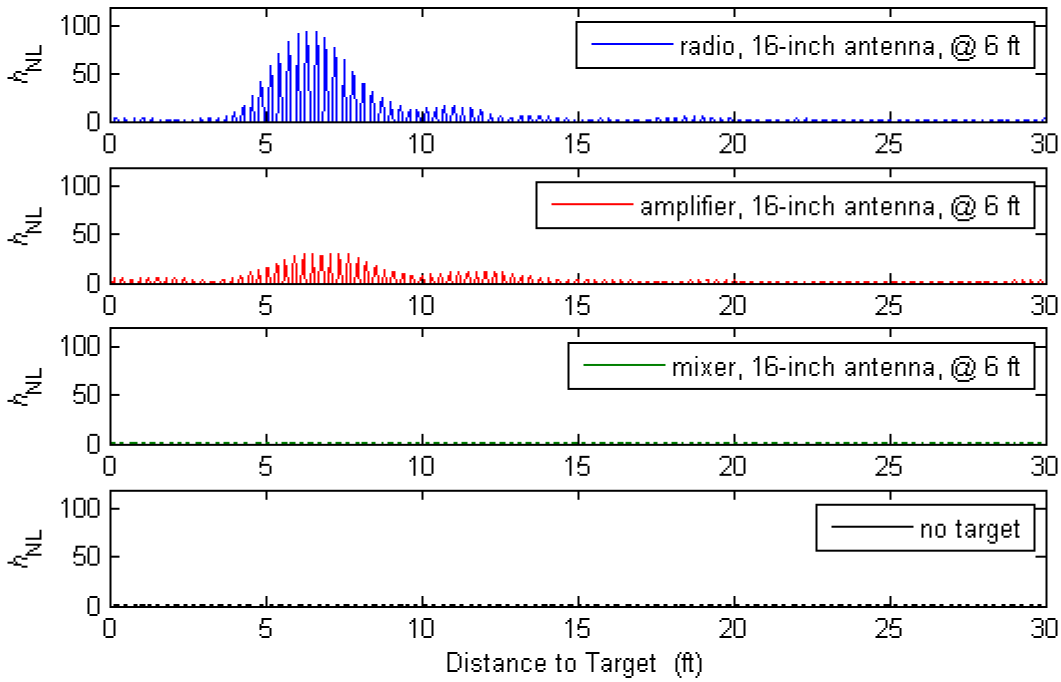

Fig. 14 Data processed for 3 targets with 16-inch antennas, placed 6 ft away from the radar antenna

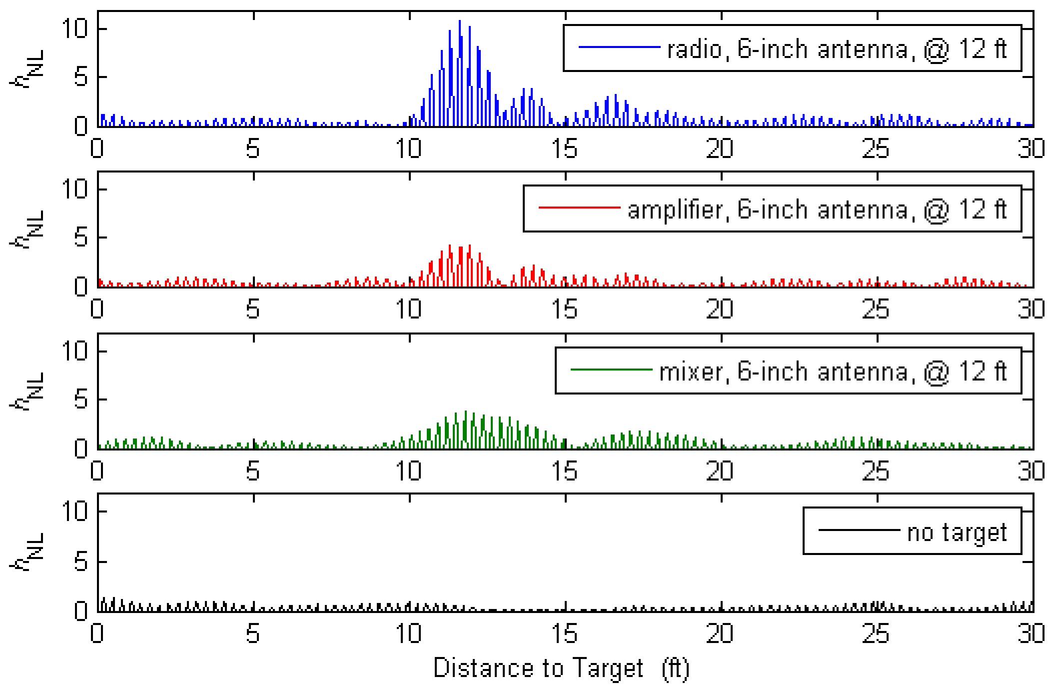

Fig. 15 Data processed for 3 targets with 6-inch antennas, placed 12 ft away from the radar antenna

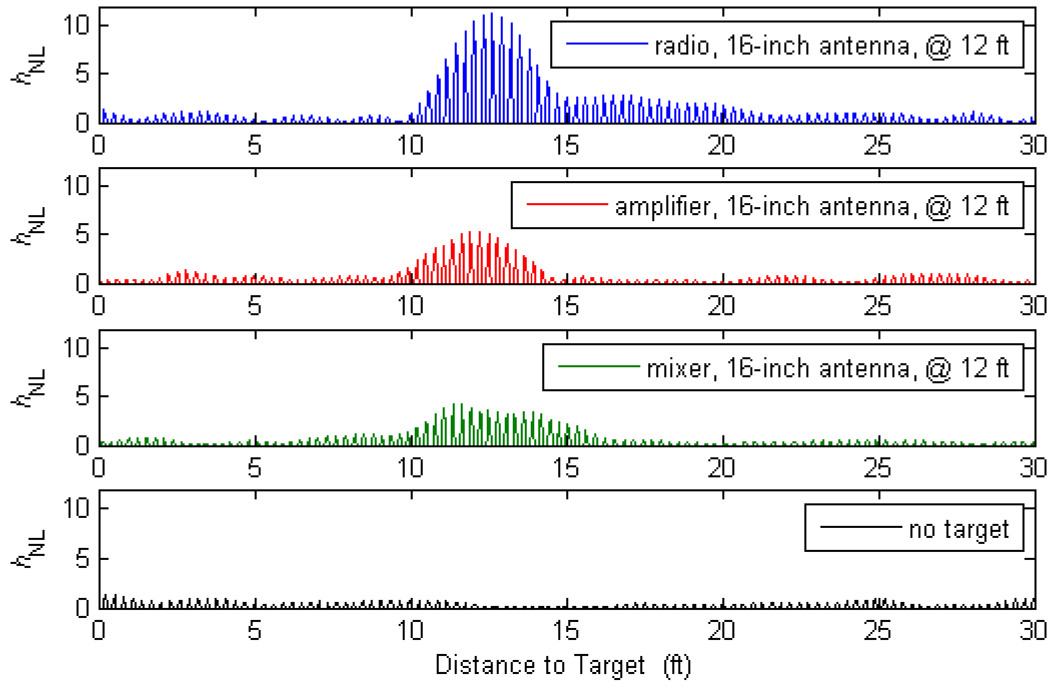

Fig. 16 Data processed for 3 targets with 16-inch antennas, placed 12 ft away from the radar antenna

First, the 6-inch antenna was connected to each target, and each target was placed on a cardboard box, 6 ft away from the antenna. The antenna was pointed directly at the target (boresight). The data recorded from the three different targets (along with a no-target, "background" case) were processed into h_{NL} and appear in Fig. 13. The 16-inch antenna was substituted for the 6-inch antenna and the same data were recorded and processed into Fig. 14. Another two data sets were recorded with the same 3 targets and 2 antennas at a distance of 12 ft away from the antenna and these data sets were processed into Figs. 15 and 16.

Since h_{NL} is nearly zero for all d in the no-target case, the system noise is minimal. Although the peaks in Figs. 13–16 are somewhat diffuse (i.e., the range resolution, the width of the main lobe of the response, is approximately 5 ft), the target responses are easily (visibly) discernible from the system noise. The drop in the peak response is approximately 20 dB from d = 6 ft to d = 12 ft, but the targets at 12 ft are still detected. Detection and ranging of the 3 nonlinear targets — using a wideband chirped waveform, harmonic detection, and computation of a nonlinear impulse response — have been demonstrated, up to a distance of 12 ft.

6. Conclusion

An experimental harmonic radar has been constructed and successfully tested at transmit frequencies between 700 to 900 MHz and an output power of 2 W. Detection of the nonlinear targets was accomplished by measuring a return signal (above the system noise) at twice the transmit frequency. Range-to-target was determined by sweeping the transmit frequencies across an ultra-wide bandwidth, capturing the amplitude and phase of the harmonic return signal, and computing the inverse Fourier transform of the wideband harmonic response. Detection and ranging of 3 nonlinear targets (radio-frequency electronic components connected to straight-wire antennas) were demonstrated, up to a distance of 12 ft away from the radar antenna.

7. References

1. Steele DW, Rotondo FS, Houck JL. Radar system for manmade device detection and discrimination from clutter. U. S. Patent 7830299, 9 Nov. 2010.

2. Kosinski JA, Palmer WD, Steer MB. Unified understanding of RF remote probing. IEEE Sensors. Dec. 2011;11(12) 3055–3063.

3. Steer MB, Wilkerson JR, Kriplani NM, Wetherington JM. Why it is so hard to find small radio frequency signals in the presence of large signals. 2012 Workshop on Integrated Nonlinear Microwave and Millimetre-Wave Circuits (INMMIC), Sept. 2012, pp. 1–3.

4. Gallagher KA, Narayanan RM, Mazzaro GJ, Sherbondy KD. Linearization of a harmonic radar transmitter by feed-forward filter reflection. Proceedings of the IEEE Radar Conference (RadarCon), Cincinnati, OH, May 2014.

5. Gallagher KA, Narayanan RM, Mazzaro GJ, Shebondy KD. Highpass filter selection for a harmonic radar receiver. conference paper in preparation, July 2014.

6. Colpitts BG, Boiteau G. Harmonic radar transceiver design: Miniature tags for insect tracking. IEEE Trans. Antennas Propag. Nov. 2004;52(11):2825–2832.

7. Psychoudakis D, Moulder W, Chen C.-C, Zhu H, Vokajus JL. A portable low-power harmonic radar system and conformal tag for insect tracking. IEEE Antennas Wireless Propag. Lett. 2008;7:444–447.

8. Singh Al, Lubecke VM. Respiratory monitoring and clutter rejection using a CW Doppler radar with passive RF tags. IEEE Sensors. Mar. 2012;12(3):558–565.

9. Charvat GL, Rothwell EJ, Kempel LC. Harmonic radar tag measurement and characterization. Proceedings of the Antennas and Propagation Society International Symposium. June 2003;2:696–699.

10. Brazee RD, Miller ES, Reding ME, Klein MG, Nudd B, Zhu H. A transponder for harmonic radar tracking of the black vine weevil in behavioral research. Trans. Amer. Soc. Agricult. Eng. Mar.–Apr. 2005;48(2):831–838.

11. O'Neial ME, Landis DA, Rothwell E, Kempel L, Reinhard D. Tracking insects with harmonic radar: A case study. Amer. Ento. Winter 2004;50(4):212–218.

12. Tahir N, Brooker G. Recent developments and recommendations for improving harmonic radar tracking systems. in Proc. 5th European Conf. Ant. Propagat. (EUCAP), Rome, Italy, pp. 1531–1535, Apr. 2011.

13. Saebboe, et al J. Harmonic automotive radar for VRU classification. Proceedings of the International Radar Conference (RADAR), pp. 1–5, Oct. 2009.

14. Mazzaro GJ, Martone AF. Multitone harmonic radar. Proceedings of the SPIE. May 2013;8714:87140E(1–7).

15. Mazzaro GJ, Martone AF, MaNamara D. M. Detection of RF electronics by multitone harmonic radar. IEEE Trans. Aerosp. Electron. Syst. Jan. 2014;50(1):477–490.

16. Laio D. A hybrid approach for characterizing linear and nonlinear electromagnetic scattering: Theory and applications. Adelphi (MD): US Army Research Laboratory (US). Report No.: ARL-TR-6261, Nov. 2012.

17. Mazzaro GJ, Gallagher KA, Martone AF, Narayanan RM. Stepped-frequency nonlinear radar simulation. Proceedings of the SPIE. May 2014;9077:90770U(1–10).

Appendix A. Measured S-Parameters for Each Component of the Harmonic Radar

Figures A-1 through A-8 show the measured S-parameters for each component of the harmonic radar .

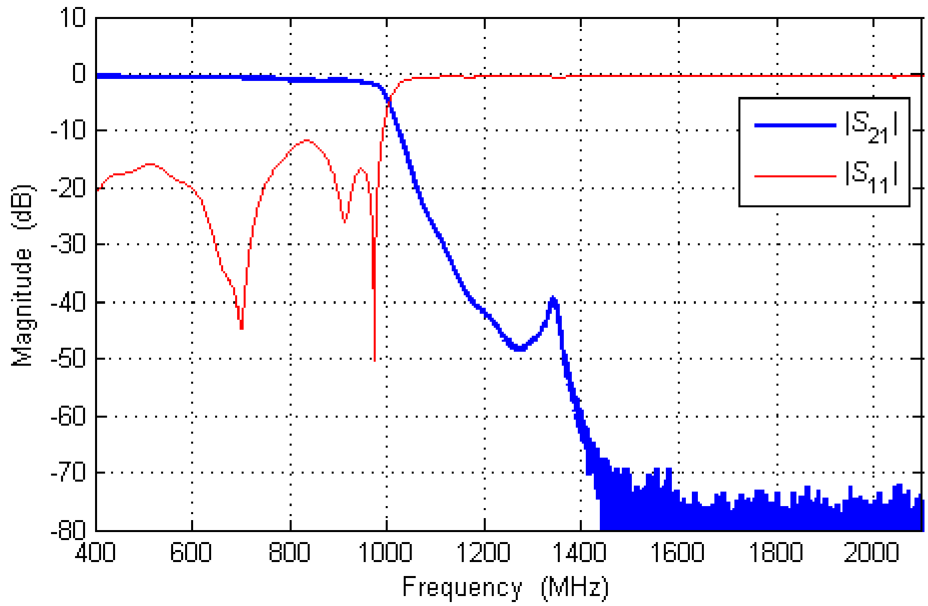

Fig. A-1 Measured S-parameters for the MiniCircuits SLP-1000+ lowpass filter pair

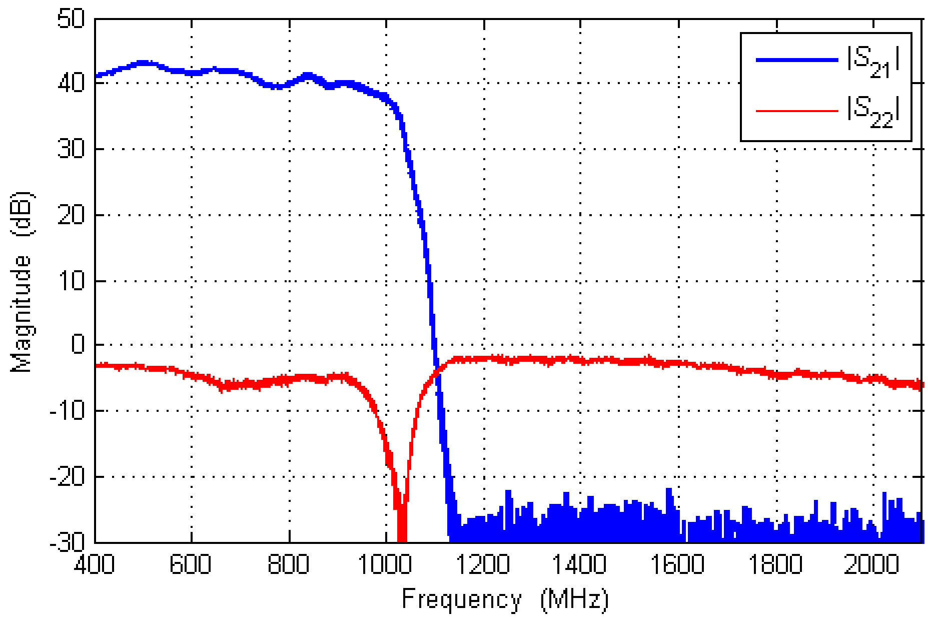

Fig. A-2 Measured S-parameters for the Amplifier Research AR4W1000 power amplifier

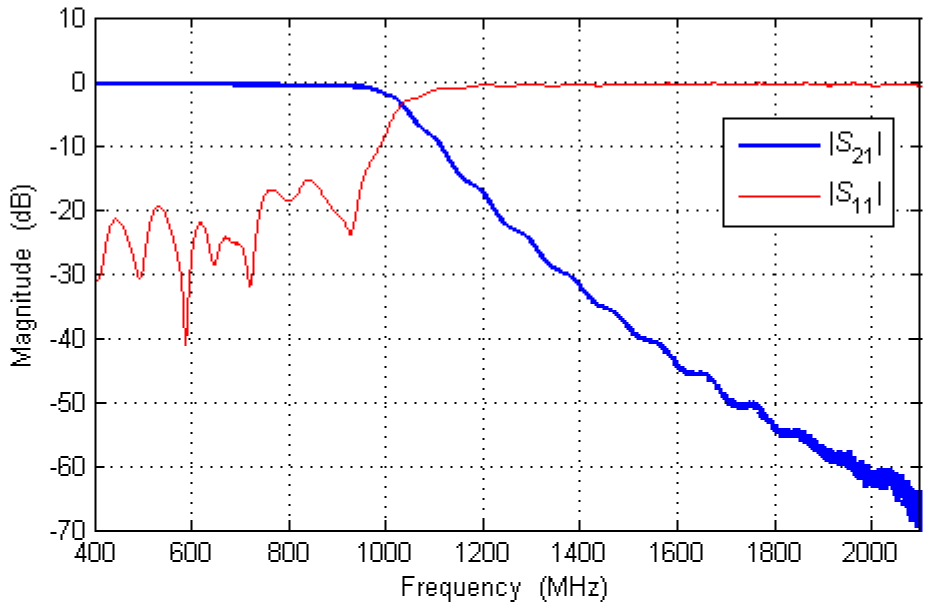

Fig. A-3 Measured S-parameters for the MiniCircuits NLP-1000+ lowpass filter

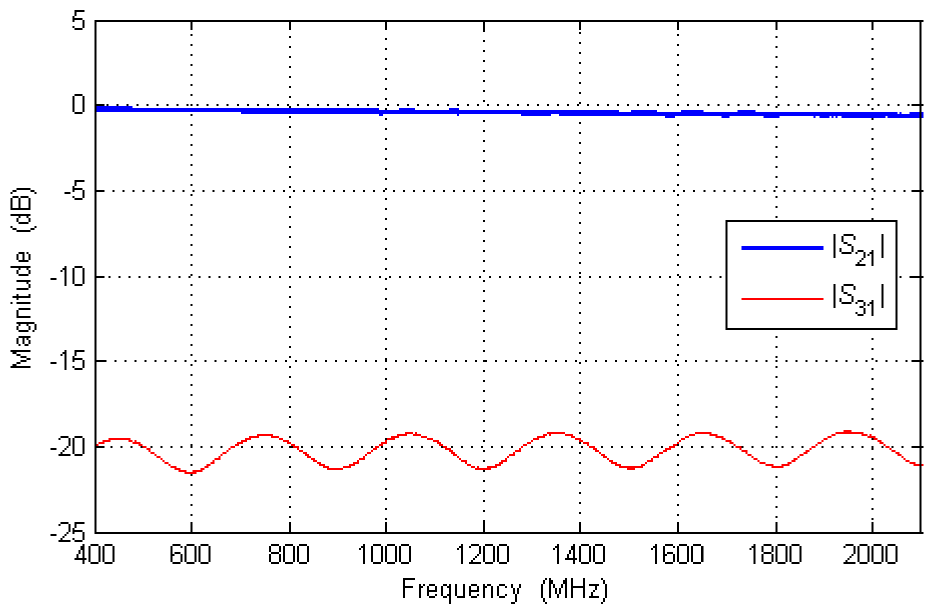

Fig. A-4 Measured S-parameters for the HP 11607-60001 directional coupler: port 1 = input, port 2 = through (output), port 3 = coupled.

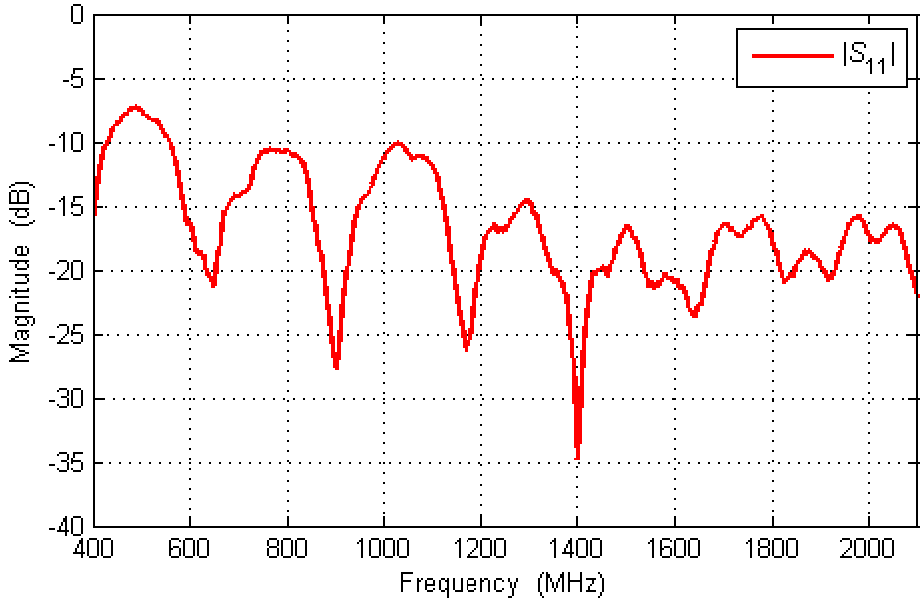

Fig. A-5 Measured S_{11} for the ETS Lindgren 3164-06 quad-ridge horn antenna

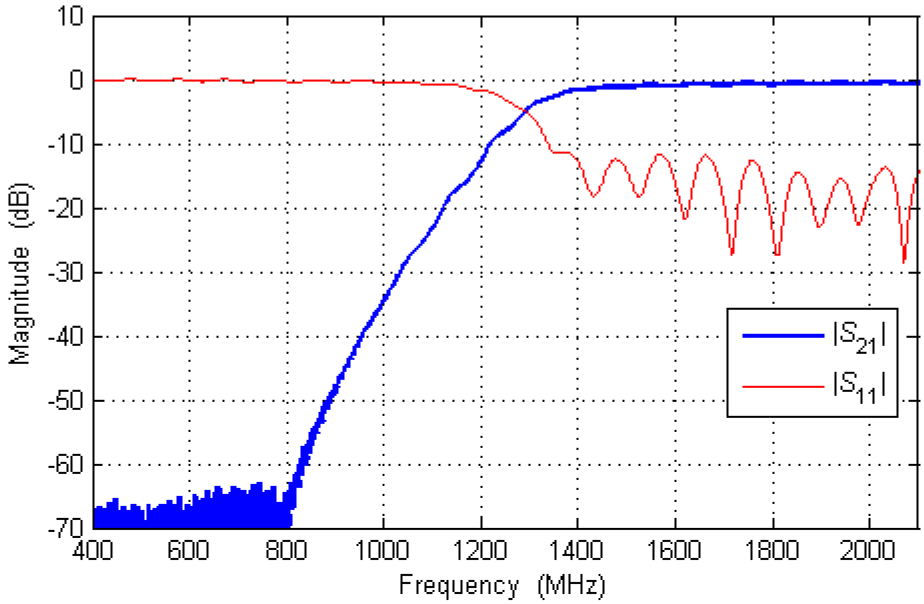

Fig. A-6 Measured S-parameters for the MiniCircuits VHF-1320+ highpass filter

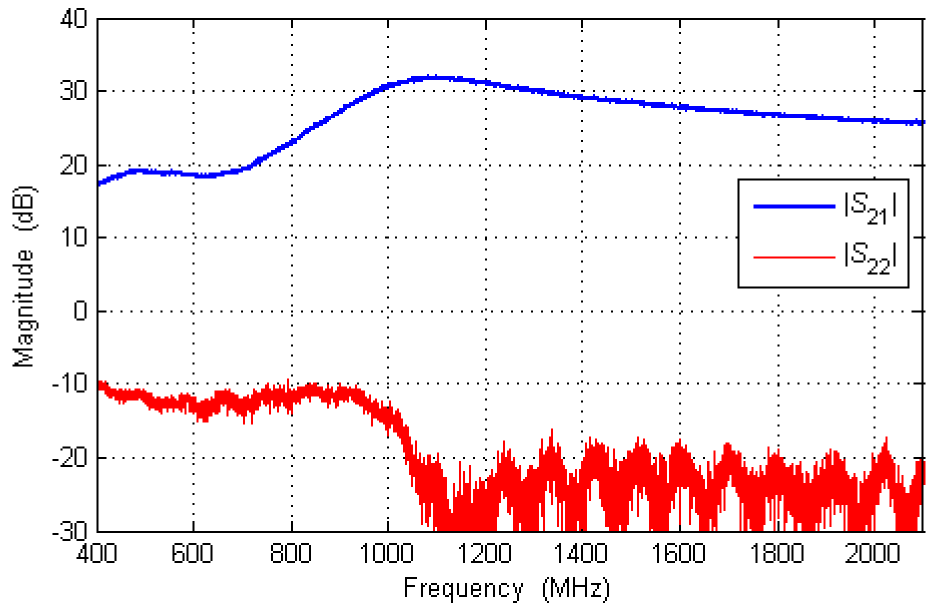

Fig. A-7 Measured S-parameters for the MiniCircuits ZRL-2400LN+ low-noise amplifier

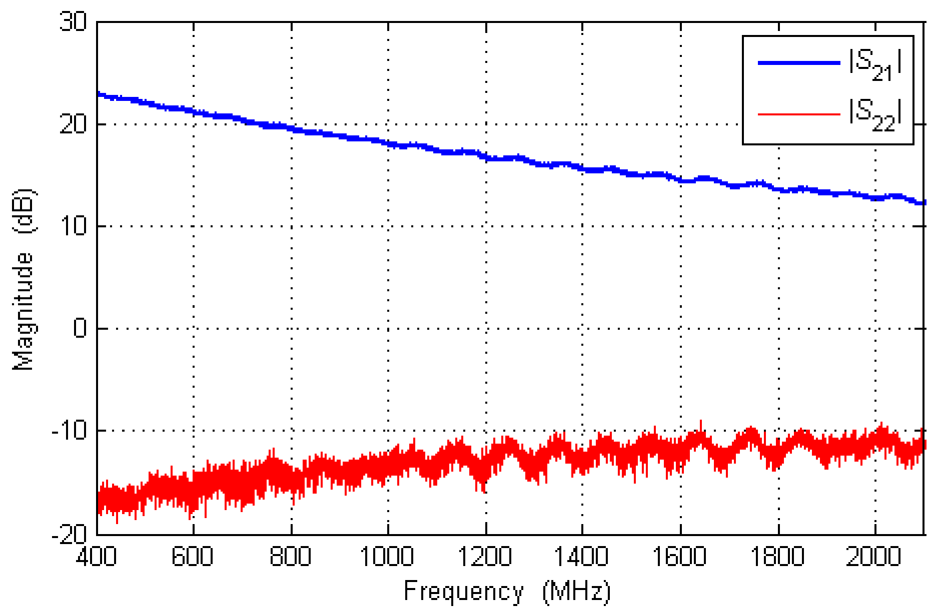

Fig. A-8 Measured S-parameters for the MiniCircuits PSA4-5043 low-noise amplifier

INTENTIONALLY LEFT BLANK.

Appendix B. Matlab Script to Control the Graphical User Interface

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% executes before the GUI is made visible %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function radar_nonlinear_OpeningFcn(hObject, eventdata, handles, varargin)

% connect to the AWG (Tektronix AWG7052)
%  and to the oscilloscope (Lecroy Wavemaster 8300A)
global awg osc;
awg = visa('ni','GPIB0::2::INSTR');
osc = visa('ni','GPIB0::6::INSTR');

fopen(awg)
fwrite(awg, 'AWGCONTROL:DOUTPUT1:STATE 1');     % turn 'direct output' on
fclose(awg)

% define strings for the Tx channel (C2), Rx channel (C3),
%  Tx channel averaged (F2), and Rx channel averaged (F3)
global data_trc_1 data_trc_2 avg_trc_1 avg_trc_2 sweeps;
data_trc_1 = 'C3';
data_trc_2 = 'C4';
avg_trc_1 = 'F3';
avg_trc_2 = 'F4';
sweeps = 1;

% turn C3 and C4 off; turn F3 and F4 on
fopen(osc)
fprintf(osc,[avg_trc_1 ':DEF EQN,''AVG(' data_trc_1 '),SUMMED'',SWEEPS,'
num2str(sweeps)]);
fprintf(osc,[avg_trc_2 ':DEF EQN,''AVG(' data_trc_2 '),SUMMED'',SWEEPS,'
num2str(sweeps)]);
fprintf(osc,[data_trc_1 ':TRACE OFF']);
fprintf(osc,[data_trc_2 ':TRACE OFF']);
fprintf(osc,[avg_trc_1 ':TRACE ON']);
fprintf(osc,[avg_trc_2 ':TRACE ON']);
fprintf(osc,[avg_trc_1 ':FRST']);           % reset the averaged sweeps
fprintf(osc,[avg_trc_2 ':FRST']);
fclose(osc)

handles.output = hObject;
guidata(hObject, handles);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% generate single-tone RF pulse %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pushbutton1_Callback(hObject, eventdata, handles)

global awg;

% user-defined waveform name to be uploaded to the AWG
global wf_name;
wf_name = get(handles.edit6,'String');
```

```
% user-defined pulse power while the pulse is active
global P_env;
P_env = str2num(get(handles.edit1,'String'));

% user-defined carrier frequency for the RF pulse
global f_pulse;
f_pulse = str2num(get(handles.edit2,'String'));

% user-defined time for the RF pulse to be active
global T_env;
T_env = str2num(get(handles.edit4,'String'));

% user-defined duty cycle
global duty_cycle;
duty_cycle = str2num(get(handles.edit5,'String'))/100;

% user-defined maximum sampling rate for the AWG
global f_sample_max;
f_sample_max = str2num(get(handles.edit3,'String'));

% generate the single-tone pulse waveform
%  and compute an appropriate AWG sampling frequency
global v fsample;
[v,f_sample] = awg_pulse(f_pulse,T_env,P_env,duty_cycle,f_sample_max);

% awg_plot(v,f_sample)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global samples;
samples = length(v);

% provide two markers (triggers) in sync with the AWG output
marker_1 = zeros(1,samples);
marker_1(1:round(samples/10)) = 1;
marker_2 = zeros(1,samples);
marker_2(1:round(samples/10)) = 1;

global buffer;
buffer = samples;
set(awg,'InputBufferSize',buffer);
set(awg,'OutputBufferSize',buffer);

% convert the real waveform data to binary
%  and upload the binary data to the AWG for playback
[binblock, header, bytes] = awg_binary(v, marker_1, marker_2);
awg_upload(awg, buffer, wf_name, binblock, header, bytes);

% set the AWG sampling frequency, select the waveform to be played,
%  and turn the AWG output on
fopen(awg);
fwrite(awg, 'AWGC:DOUT1 ON');
fwrite(awg, ['SOURCE1:FREQUENCY ' num2str(f_sample/10^9) ' GHZ']);
```

```
fwrite(awg, ['SOURCE1:WAVEFORM "' wf_name '"']);
fwrite(awg, 'OUTPUT ON');
fwrite(awg, 'AWGCONTROL:RUN');
fclose(awg);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% generate linear RF chirp pulse %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pushbutton2_Callback(hObject, eventdata, handles)

global awg;

% user-defined waveform name to be uploaded to the AWG
global wf_name;
wf_name = get(handles.edit12,'String');

% user-defined chirp power while the pulse is active
global P_env;
P_env = str2num(get(handles.edit8,'String'));

% user-defined start frequency for the chirp
global f_start;
f_start = str2num(get(handles.edit9,'String'));

% user-defined end frequency for the chirp
global f_end;
f_end = str2num(get(handles.edit13,'String'));

% user-defined length of the chirp
global T_env;
T_env = str2num(get(handles.edit10,'String'));

% user-defined duty cycle
global duty_cycle;
duty_cycle = str2num(get(handles.edit11,'String'))/100;

% user-defined maximum sampling rate for the AWG
global f_sample_max;
f_sample_max = str2num(get(handles.edit3,'String'));

% generate the RF chirp waveform
%  and compute an appropriate AWG sampling frequency
global v f_sample;
[v,f_sample] = awg_chirp(f_start,f_end,T_env,P_env,duty_cycle,f_sample_max);

% awg_plot(v,f_sample)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global samples;
samples = length(v);
```

```
% provide two markers (triggers) in sync with the AWG output
marker_1 = zeros(1,samples);
marker_1(1:round(samples/20)) = 1;
marker_2 = zeros(1,samples);
marker_2(1:round(samples/20)) = 1;

global buffer;
buffer = samples;
set(awg,'InputBufferSize',buffer);
set(awg,'OutputBufferSize',buffer);

% convert the real waveform data to binary
%  and upload the binary data to the AWG for playback
[binblock, header, bytes] = awg_binary(v, marker_1, marker_2);
awg_upload(awg, buffer, wf_name, binblock, header, bytes);

% set the AWG sampling frequency, select the waveform to be played,
%  and turn the AWG output on
fopen(awg);
fwrite(awg, 'AWGC:DOUT1 ON');
fwrite(awg, ['SOURCE1:FREQUENCY ' num2str(f_sample/10^9) ' GHZ']);
fwrite(awg, ['SOURCE1:WAVEFORM "' wf_name '"']);
fwrite(awg, 'OUTPUT ON');
fwrite(awg, 'AWGCONTROL:RUN');
fclose(awg);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% generate multitone RF pulse %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pushbutton4_Callback(hObject, eventdata, handles)

global awg;

% user-defined waveform name to be uploaded to the AWG
global wf_name;
wf_name = get(handles.edit18,'String');

global N;
N = str2num(get(handles.edit19,'String'));

% user-defined power per tone while the pulse is active
global P_tone;
P_tone = str2num(get(handles.edit14,'String'));

% user-defined center frequency for the tones
global f_c;
f_c = str2num(get(handles.edit15,'String'));

% user-defined time for the multitone pulse to be active
global T_env;
T_env = str2num(get(handles.edit16,'String'));
```

```
% user-defined duty cycle
global duty_cycle;
duty_cycle = str2num(get(handles.edit17,'String'))/100;

% user-defined maximum sampling rate for the AWG
global f_sample_max;
f_sample_max = str2num(get(handles.edit3,'String'));

% generate the multitone waveform
%  and compute an appropriate AWG sampling frequency
global v fsample;
[v,f_sample] = awg_multitone(N,f_c,T_env,P_tone,duty_cycle,f_sample_max);

% awg_plot(v,f_sample)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global samples;
samples = length(v);

% provide two markers (triggers) in sync with the AWG output
marker_1 = zeros(1,samples);
marker_1(1:round(samples/10)) = 1;
marker_2 = zeros(1,samples);
marker_2(1:round(samples/10)) = 1;

global buffer;
buffer = samples;
set(awg,'InputBufferSize',buffer);
set(awg,'OutputBufferSize',buffer);

% convert the real waveform data to binary
%  and upload the binary data to the AWG for playback
[binblock, header, bytes] = awg_binary(v, marker_1, marker_2);
awg_upload(awg, buffer, wf_name, binblock, header, bytes);

% set the AWG sampling frequency, select the waveform to be played,
%  and turn the AWG output on
fopen(awg);
fwrite(awg, 'AWGC:DOUT1 ON');
fwrite(awg, ['SOURCE1:FREQUENCY ' num2str(f_sample/10^9) ' GHZ']);
fwrite(awg, ['SOURCE1:WAVEFORM "' wf_name '"']);
fwrite(awg, 'OUTPUT ON');
fwrite(awg, 'AWGCONTROL:RUN');
fclose(awg);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% generate stepped-frequency pulse %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pushbutton5_Callback(hObject, eventdata, handles)

global awg;
```

```
% user-defined waveform name to be uploaded to the AWG
global wf_name;
wf_name = get(handles.edit24,'String');

% user-defined power at each frequency while the pulse is active
global P_env;
P_env = str2num(get(handles.edit20,'String'));

% user-defined start frequency for the steps
global f_start;
f_start = str2num(get(handles.edit26,'String'));

% user-defined end frequency for the steps
global f_end;
f_end = str2num(get(handles.edit21,'String'));

% user-defined frequency step size
global f_delta;
f_delta = str2num(get(handles.edit25,'String'));

% user-defined time for the stepped waveform to be active
global T_env;
T_env = str2num(get(handles.edit22,'String'));

% user-defined duty cycle
global duty_cycle;
duty_cycle = str2num(get(handles.edit23,'String'))/100;

% user-defined maximum sampling rate for the AWG
global f_sample_max;
f_sample_max = str2num(get(handles.edit3,'String'));

% generate the stepped-frequency waveform
%  and compute an appropriate AWG sampling frequency
global v f_sample;
[v,f_sample] =
awg_stepped(f_start,f_end,f_delta,T_env,P_env,duty_cycle,f_sample_max);

% awg_plot(v,f_sample)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global samples;
samples = length(v);

% provide two markers (triggers) in sync with the AWG output
marker_1 = zeros(1,samples);
marker_1(1:round(samples/2)) = 1;
marker_2 = zeros(1,samples);
marker_2(1:round(samples/2)) = 1;

global buffer;
buffer = samples;
```

```
set(awg,'InputBufferSize',buffer);
set(awg,'OutputBufferSize',buffer);

% convert the real waveform data to binary
%  and upload the binary data to the AWG for playback
[binblock, header, bytes] = awg_binary(v, marker_1, marker_2);
awg_upload(awg, buffer, wf_name, binblock, header, bytes);

% set the AWG sampling frequency, select the waveform to be played,
%  and turn the AWG output on
fopen(awg);
fwrite(awg, 'AWGC:DOUT1 ON');
fwrite(awg, ['SOURCE1:FREQUENCY ' num2str(f_sample/10^9) ' GHZ']);
fwrite(awg, ['SOURCE1:WAVEFORM "' wf_name '"']);
fwrite(awg, 'OUTPUT ON');
fwrite(awg, 'AWGCONTROL:RUN');
fclose(awg);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% clear waveforms & reset AWG %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pushbutton3_Callback(hObject, eventdata, handles)

global awg;

fopen(awg);
fwrite(awg,'*RST');
fwrite(awg,'*CLS');
pause(1)
fwrite(awg, 'AWGCONTROL:DOUTPUT1:STATE 1');
fclose(awg);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% capture and process Rx waveform %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pushbutton7_Callback(hObject, eventdata, handles)

global awg osc;

global trig_src;
switch get(handles.popupmenu2,'Value')
    case 1
        trig_src = 'EX';                    % trigger = external
    case 2
        trig_src = 'C1';                    % trigger = channel C1
    case 3
        trig_src = 'C2';                    % trigger = channel C4
    otherwise
        trig_src = 'EX';
end
```

```
% user-defined trigger level
global trig_lev;
trig_lev = str2num(get(handles.edit37,'String'))*10^-3;

% user-defined total time for Tx and Rx data to be collected
global t_total;
t_total = str2num(get(handles.edit32,'String'))*10^-6;
t_div = t_total/10;

% user-defined voltage per division visible on the oscilloscope
global v_div_1 v_div_2;
v_div_1 = str2num(get(handles.edit33,'String'))*10^-3;
v_div_2 = str2num(get(handles.edit34,'String'))*10^-3;

% user-defined voltage offset on the oscilloscope
global v_offset;
v_offset = 0e-3;

% set trigger delay to 1/2 of the data collection time so that
%  the maximum amount of data is collected AFTER the trigger point
global trig_delay;
trig_delay = -t_total/2;

% user-defined Matlab file to store Tx and Rx time & voltage vectors
global data_name;
data_name = get(handles.edit36,'String');

global trig_mode;
trig_mode = 'SINGLE';           % use a single trigger for each sweep

global f_capture;
f_capture = 10e9;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global buffer;
buffer = 10*t_div*f_capture + 26;
set(osc,'InputBufferSize',buffer);
set(osc,'OutputBufferSize',buffer);

global data_trc_1 data_trc_2 avg_trc_1 avg_trc_2 sweeps;
sweeps = str2num(get(handles.edit38,'String'));

% set the oscilloscope to record data according to user-input values
fopen(osc);
fprintf(osc, ['TRSE EDGE,SR,' trig_src ',HT,OFF']);
fprintf(osc, ['TRMD ' trig_mode]);
fprintf(osc, ['TRDL ' num2str(trig_delay/10^-6) ' US']);
fprintf(osc, ['EX:TRLV ' num2str(trig_lev/10^-3) ' MV']);
fprintf(osc, ['TDIV ' num2str(t_div/10^-6) ' US']);
fprintf(osc, [data_trc_1 ':VDIV ' num2str(v_div_1/10^-3) ' MV']);
fprintf(osc, [data_trc_2 ':VDIV ' num2str(v_div_2/10^-3) ' MV']);
fprintf(osc, [data_trc_1 ':OFST ' num2str(v_offset/10^-3) ' MV']);
```

```
fprintf(osc, [data_trc_2 ':OFST ' num2str(v_offset/10^-3) ' MV']);
fclose(osc);

% turn the appropriate data traces on/off
%  and set two of the traces to each average the Tx or Rx data stream
fopen(osc);
fprintf(osc,[data_trc_1 ':TRACE OFF']);
fprintf(osc,[data_trc_2 ':TRACE OFF']);
fprintf(osc,[avg_trc_1 ':TRACE ON']);
fprintf(osc,[avg_trc_2 ':TRACE ON']);
fprintf(osc,[avg_trc_1 ':FRST']);
fprintf(osc,[avg_trc_2 ':FRST']);
fprintf(osc,[avg_trc_1 ':DEF EQN,''AVG(' data_trc_1 '),SUMMED'',SWEEPS,'
num2str(sweeps)]);
fprintf(osc,[avg_trc_2 ':DEF EQN,''AVG(' data_trc_2 '),SUMMED'',SWEEPS,'
num2str(sweeps)]);
for counter = 1:sweeps
    fprintf(osc,'*TRG');
    fprintf(osc,'WAIT');
end
fclose(osc);

% download the oscilloscope data into Matlab
[v_trans, v_rec, t_trans, t_rec] = osc_capture( osc, buffer, avg_trc_1,
avg_trc_2 );

% stop the waveform generator playback
fopen(awg);
fwrite(awg, 'OUTPUT OFF');
fwrite(awg, 'AWGCONTROL:STOP');
fclose(awg);

% save the raw Tx and Rx vectors to a MAT file
save(data_name,'t_trans','v_trans','t_rec','v_rec');

% process and display the Tx and Rx data
osc_process( v_trans, v_rec, t_trans, t_rec );
```

Appendix C. Matlab Function to Generate a Chirp Waveform

```
function [v, f_sample] = awg_chirp( f_start, f_end, T_env, P_env, duty_cycle,
f_sample_max )

f_res = 1/T_env;                             % frequency resolution (Hz)

f_sample = 2*f_res;
while (f_sample <= f_sample_max/2)           % adjust AWG sample rate
    f_sample = 2*f_sample;
end

f_start = round(f_start/f_res)*f_res;        % adjust start frequency
f_end = round(f_end/f_res)*f_res;            % adjust end frequency

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t_sbb = 1/f_sample;                          % sampling time
t = 0:t_sbb:T_env-t_sbb;                     % time vector
A = sqrt(10^(P_env/10)*2*50*10^-3);          % amplitude of chirp when ON

k = (f_end-f_start)/T_env;                   % chirp rate (Hz/s)

x = A * cos(2*pi*(f_start+(k/2)*t).*t);      % chirp equation

if (duty_cycle <= 0) || (duty_cycle >= 1)
    v = x;                                   % continuous wave
else
    v = [x zeros(1,(round(1/duty_cycle)-1)*length(x))];     % pulsed
end
```

Appendix D. Matlab Function to Generate a Step-Frequency Waveform

```
function [v, f_sample] = awg_stepped( f_start, f_end, f_delta, T_env, P_env,
duty_cycle, f_sample_max )

f_res = 1/T_env;                              % frequency resolution (Hz)

f_sample = 2*f_res;
while (f_sample <= f_sample_max/2)            % adjust AWG sample rate
    f_sample = 2*f_sample;
end

freq = (f_start:f_delta:f_end-f_delta);     % instantanous frequencies
N = length(freq);

% adjust instantaneous frequencies
for counter = 1:length(freq)
    freq(counter) = floor(freq(counter)/f_res)*f_res;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t_sbb = 1/f_sample;                           % sampling time
t = 0:t_sbb:T_env-t_sbb;                      % time vector
A = sqrt(10^(P_env/10)*2*50*10^-3);           % amplitude of chirp when ON

index = ceil((1:length(t))/(length(t)/N));
f = freq(index);
x = A * cos(2*pi*(f.*t));

if (duty_cycle <= 0) || (duty_cycle >= 1)
    v = x;                                    % continuous wave
else
    v = [x zeros(1,(round(1/duty_cycle)-1)*length(x))];     % pulsed
end
```

Appendix E. Matlab Function to Process the Received Waveform into h_{NL}

```
function osc_process( v_1, v_2, t_1, t_2 )

v_in = v_1;
v_ref = v_in.^2;
v_out = v_2;

t = (0:1:length(v_in)-1) * 1e-10;
v_out = bandpass( t, v_out, 700e6*2, 900e6*2 );

delta_t = t(2)-t(1);
f_sample = 1/delta_t;
T = t(length(t)) + 1/f_sample;
freq = -f_sample/2:1/T:(f_sample/2)-1/T;

V_in  = fftshift(fft(2*(v_in))/length(v_in));
V_ref  = fftshift(fft(2*(v_ref))/length(v_ref));
V_out  = fftshift(fft(2*(v_out))/length(v_out));

H = V_out./V_ref;
h = ifft(fftshift(H/2)*length(H));

% adjust the velocity of propagation by dielectric constant
dielectric_PTFE = 2.1;
velocity_PTFE = 0.9836e9 / sqrt(dielectric_PTFE);
d = velocity_PTFE * t/2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(10)
plot(d,real(h),'k')
axis([0 40 0 Inf])
ylabel('{\ith}_{NL}  (rescaled)')
xlabel('Distance to Target  (ft)')
```

1	DEFENSE TECH INFO CTR
(PDF)	ATTN DTIC OCA
2	US ARMY RSRCH LABORATORY
(PDF)	ATTN IMAL HRA MAIL & RECORDS MGMT
	ATTN RDRL CIO LL TECHL LIB
1	GOVT PRNTG OFC
(PDF)	ATTN A MALHOTRA
5	US ARMY RSRCH LAB
(PDF)	ATTN RDRL SER U
	GREGORY J MAZZARO
	KYLE A GALLAGHER
	ALBERT R OWENS
	KELLY D SHERBONDY
	RAM M NARAYANAN

INTENTIONALLY LEFT BLANK.